Tutorial: Hello World 3

Hello World shows you how to put together an application with the THINK Project Manager. The idea here is not to write a fancy program, but to show you how to build an application in the THINK Project Manager. The program writes the words “hello world!” in a window on the screen.

Before you begin

Be sure you followed the instructions in Chapter 2, “Installation,” to install the THINK Project Manager on your disk.

What you should know

You should know how to use the standard file dialog boxes to move to different folders. If you don’t know how to do this, read the user’s manual that came with your Macintosh.

Contents

Creating the Project 37

Creating the Source File 41

Compiling the Source File 42
Did you get an error? 43

Adding the Library. 44

Running the Project 46

Creating the Application 48

Where to Go Next 49

3 Tutorial: Hello World

Creating the Project

First you need to launch the THINK Project Manager. To do this, open the folder containing the THINK Project Manager and double-click the THINK Project Manager icon.

You see a dialog box that asks you to open a project. (If you have Symantec C++, you see a different folder name.)

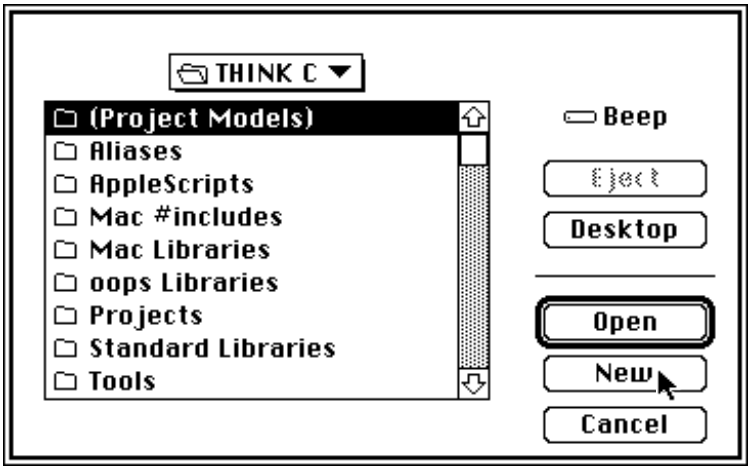

Figure 3-1 Opening a project

Since you're creating a new project, click the New button. You see another dialog box, one that lets you select different types of projects to create.

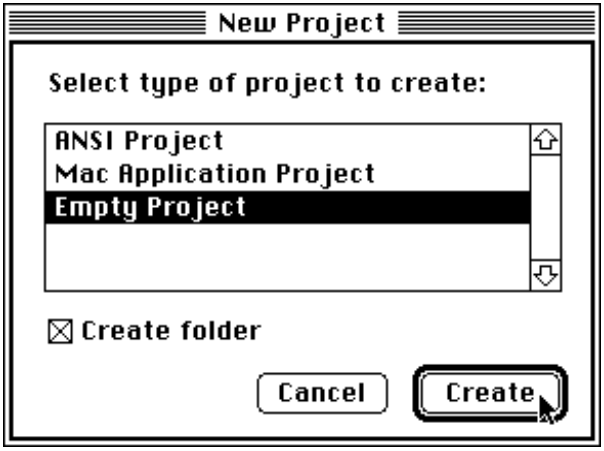

Figure 3-2 Choosing the type of project to create

Since this tutorial teaches you how to build a simple project from scratch, select Empty Project from the list and click the Create button. Generally, when you create a new project, you use one of the built-in project models or a custom project model that you designed.

Now you see a dialog asking you to name your new project and choose where to store it. Move back to the `Development` folder, name the project `hello`, and click the Save button.

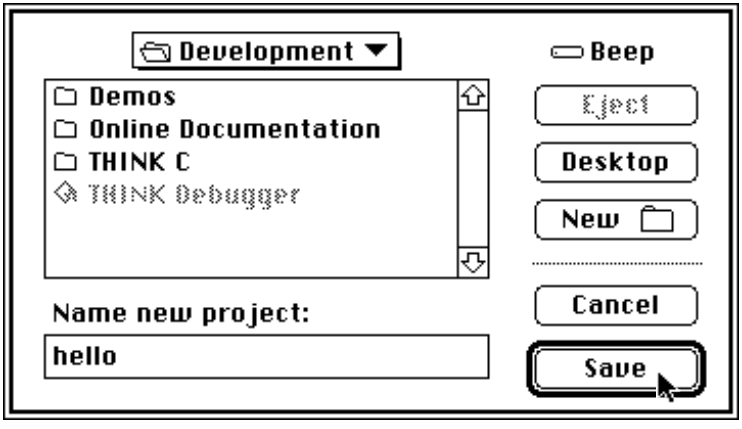

Figure 3-3 Naming your new project

You can give your project a different name, or choose to store it somewhere other than the Development folder, but remember that your dialog boxes won't match the pictures in this chapter.

Note

Generally, you don't store projects in the THINK Tree (the folder where the THINK Project Manager and its subfolders reside). In particular, despite its name, you don't store projects in the Projects folder. This folder is for storing aliases to frequently used projects that you want to appear in the **Switch to Project** hierarchical menu.

The THINK Project Manager creates a new folder on disk named hello *f*, and inside it a new project document named hello.π. Then the THINK Project Manager displays the new, empty project window:

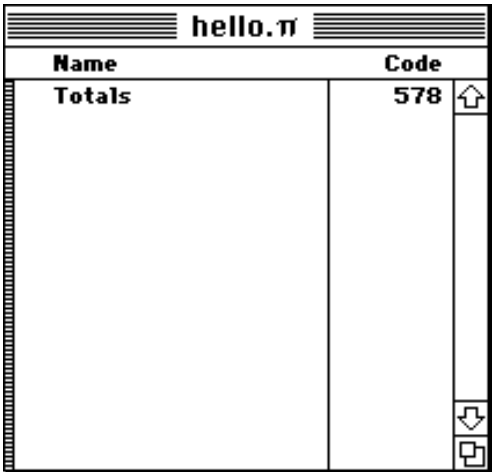

Figure 3-4 Project window for a new project file

The Name column shows the names of all the source files and libraries in your project, and the Code column displays their sizes in bytes.

If you're running under System 7 (or MultiFinder under System 6), go back to the Finder and look at the Development folder. Notice that

the THINK Project Manager created a folder for you named `hello ƒ`, and placed a file within it: `hello.π`, the project document.

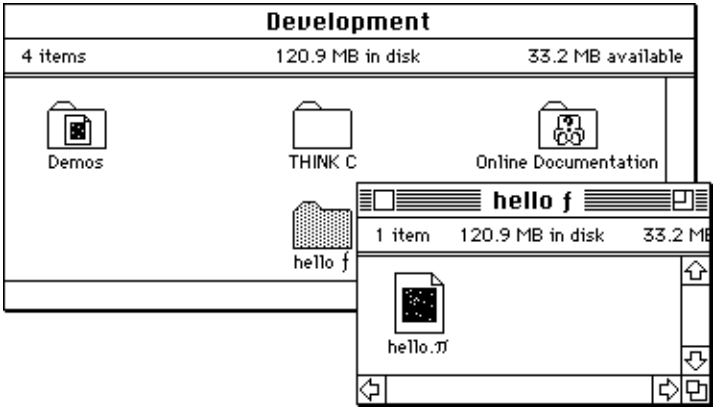

Figure 3-5 hello ƒ folder

Creating the Source File

Now you're ready to create your source file. Choose **New** from the **File** menu to bring up an empty edit window.

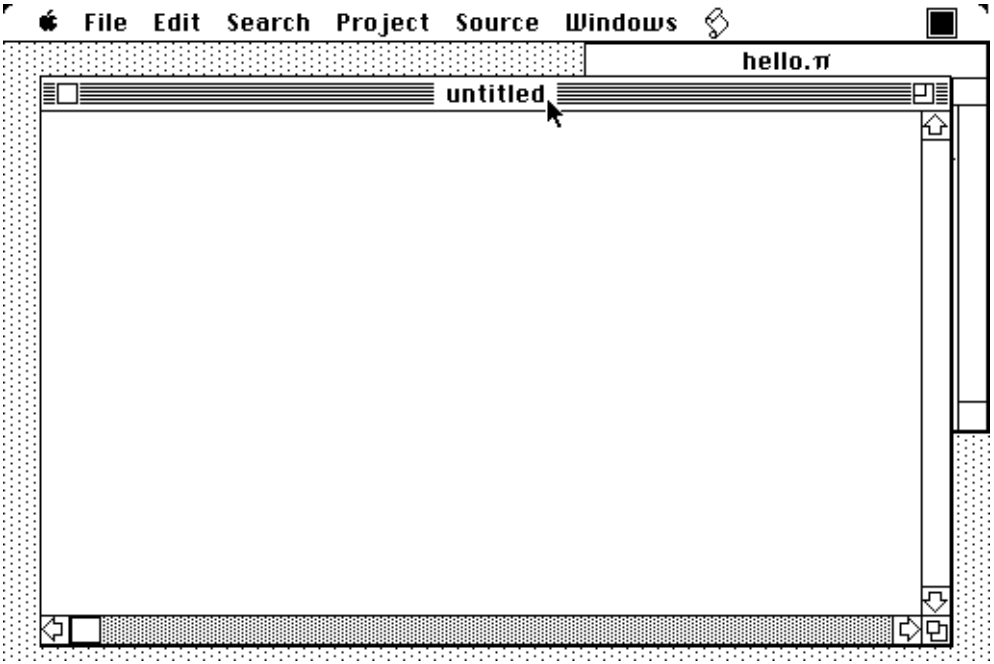

Figure 3-6 Empty edit window

Type this program into the edit window (you don't need to type in the comments if you're in a hurry):

```
/*****
 * hello.c
 *
 * The hello world program for
 * the THINK Project Manager
 *
 *****/

#include <stdio.h>

main()
{
      printf("hello world!\n");
}
```

The THINK Project Manager's text editor works like most other text editors on the Macintosh. You can drag to select a range of text or double-click to select words. You can also triple-click to select an entire line. If you have a keyboard with arrow keys, you can use them to move around your file.

For more information about the THINK Project Manager text editor, see Chapter 10, "Editing Files."

The text editor does not wrap text back to the left edge of the window when you type past the right edge of the window. Use the horizontal scroll bar at the bottom of the window to see any text that goes past the right edge.

When you've typed in the program, choose **Save As** from the **File** menu to save it. You'll get a dialog box like the one below. Name the file `hello.c`, and click the Save button.

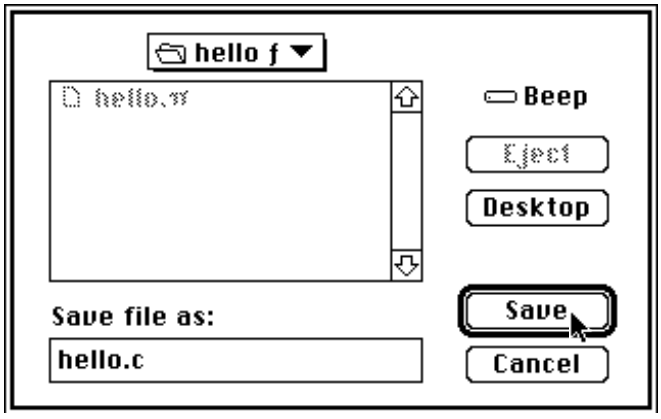

Figure 3-7 Saving a new file

You can edit any text file with the THINK Project Manager's editor, not just those ending in `.c`.

Compiling the Source File

*Using the **Compile** command in the THINK Project Manager is like using the* `cc` *command in Unix, except that the THINK Project Manager adds the object code to your project, instead of creating a separate object file.*

Now you're ready to compile your source file. Choose **Compile** from the **Source** menu. The THINK Project Manager displays a dialog box that shows how many lines have been compiled.

When the THINK Project Manager compiles a source file, it adds its name and size to the project window. Your project window should now look like this:

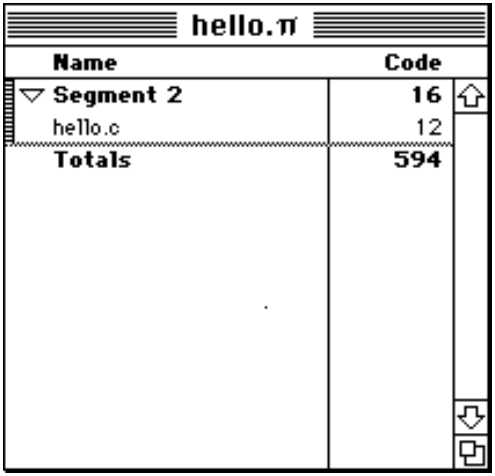

Figure 3-8 Project window with one file

The THINK Project Manager keeps all the object code for your source files in the project document.

Did you get an error?

If you made a mistake typing the program, the THINK Project Manager displays an error message in a dialog box. The message may say `syntax error`. In this small program, about the only syntax error you can make is forgetting a quote, a parenthesis, or a semicolon.

Click anywhere in the dialog box to get rid of it. The THINK Project Manager puts the insertion point in the line with the error. Look over your program to make sure everything is correct. Then choose **Compile** from the **Source** menu.

If you get an error message that says `can't open file`:

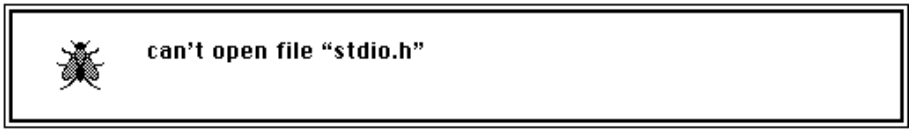

Figure 3-9 Error message

it means that the THINK Project Manager wasn't able to find the `#include` file `stdio.h`. The THINK Project Manager can't find the `#include` files if you didn't move the `Standard Libraries` folder into the THINK Project Manager folder. If that is the case, the best thing to do is to start over from the beginning.

Quit the THINK Project Manager and move the `hello ƒ` folder to the Trash. Then look in Chapter 2, "Installation," to make sure you installed the THINK Project Manager correctly. Once you're sure everything is OK, start again from the beginning of this chapter.

Warning

Throw the `hello ƒ` folder into the Trash only if you're starting all over. If you didn't get the `can't open file` error message, go on.

Adding the Library

If you tried to run your program now, you'd get linking errors because the project doesn't know where the `printf()` function is defined.

To learn more about the routines in the ANSI library, see the Online Standard Libraries Reference.

Next, you need to add the ANSI library to your project. It defines the routine `printf()` and all the standard C library routines. To add

the ANSI library, choose **Add Files** from the **Source** menu. You see a dialog like the one in Figure 3-10.

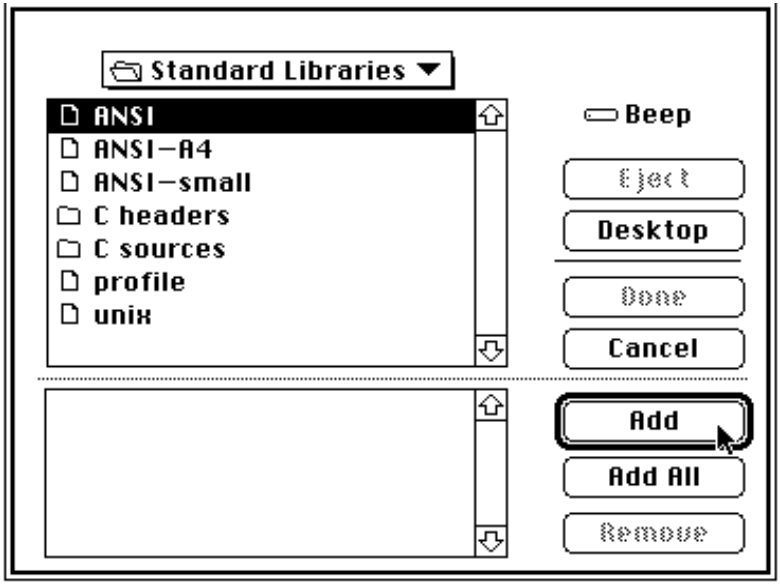

Figure 3-10 Adding a dialog

The top list displays the source files and libraries in the folder you're currently in. The bottom list shows the files the THINK Project Manager will add to your project when you click Done. The Add and Add All buttons move files into the bottom list. The Remove button removes files from the bottom list.

Move to the folder called `Standard Librarie`s. This folder contains all the libraries for ANSI and Unix compatibility, including the ANSI library. Select ANSI and click the Add button. The file moves to the bottom list. Now click the Done button.

The THINK Project Manager adds the name `ANSI` to the project window. Your project window should look like this:

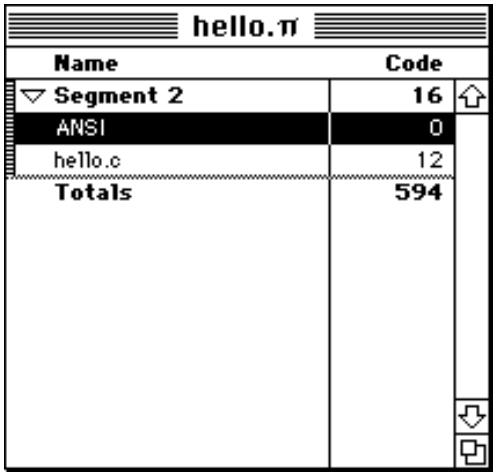

Figure 3-11 hello project window with both its files

The object size for the ANSI library is zero since the THINK Project Manager doesn't load a library's code until you need it. This lets you add several libraries without waiting for them to load.

The THINK Project Manager loads a library automatically when you run the project. Another way to load a library is to click its name in the project window, and then choose **Compile** from the **Source** menu. For this example, let the THINK Project Manager load it for you.

Running the Project

*Using the **Run** command in the THINK Project Manager is like doing all this in most other compilers: compiling your project's files, linking, and running the project. However, **Run** doesn't save a copy of your application to disk before running it. To save it, use **Build Application**, described below.*

Everything is all set to run the project. The source file is in the project window along with the libraries you'll be using. Now choose **Run** from the **Project** menu.

The THINK Project Manager notices that the library needs to be loaded, so it puts up a dialog box asking you if you want to bring the project up to date:

Figure 3-12 Dialog box asking if you want to update your project

Click the Update button. The THINK Project Manager goes to disk to load the code for the ANSI library. It may take the THINK Project Manager some time to load it. Once it's loaded into the project, though, the THINK Project Manager doesn't need to load it again.

Any time you choose to run your project, and the THINK Project Manager notices that you've made changes (added libraries or source files or edited source files), it asks you if you want to update the project. If you say yes, it compiles the new or changed files and loads the new libraries.

For more information on console windows, see the Online Standard Libraries Reference.

Since this program uses the ANSI library, everything `printf()` writes goes to a console window. A console window is a Macintosh window that behaves like a simple display terminal, the kind of

terminal that many MS-DOS and Unix computers use. You see the "hello world!" string at the bottom of this window.

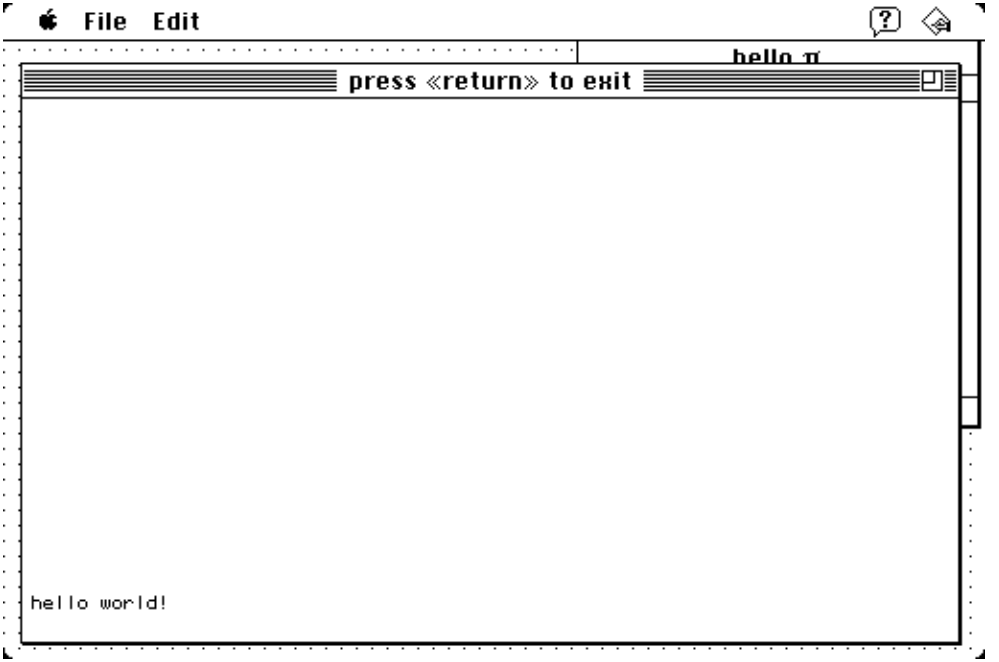

Figure 3-13 Your program running in a console window

To exit the program, press the Return key or choose **Quit** from the **File** menu.

Creating the Application

As you develop a large application, you make changes to your source files. Each time you run your project, the THINK Project Manager recompiles only those files that have changed. When you're ready to turn your project into a stand-alone double-clickable application, choose **Build Application** from the **Project** menu.

You see a dialog box asking you to name your application. Name it `hello appl`. Leave the Smart Link box checked. This option tells

the THINK Project Manager to make your application as small as possible.

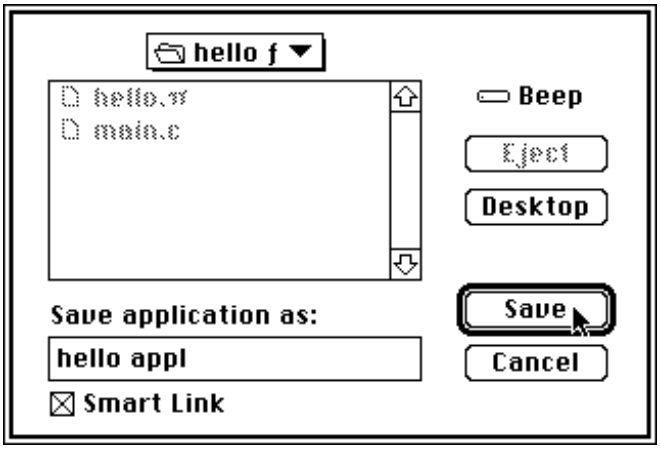

Figure 3-14 Building your application

The THINK Project Manager puts up a dialog box telling you it's linking your application. When it's finished, the application is in the folder you chose.

If you're not using System 7 (or MultiFinder under System 6), quit the THINK Project Manager to run your application. Use the **Quit** command in the **File** menu. If you're using System 7 (or MultiFinder under System 6), you don't need to quit first. Just bring up the window with the folder your application is in. Double-click your application and watch it run. That's all there is to it.

Where to Go Next

The tutorial in the next chapter is a more elaborate example of building an application with the THINK Project Manager. It describes how the THINK Project Manager reports errors when you compile and link, and it shows you some advanced features of the THINK Project Manager's editor.

If you would rather explore on your own, read the chapters that interest you in Part Three of this guide, "Using the THINK Project Manager." Or, if you want to learn how to use the THINK Project Manager's source-level debugger, go to Chapter 5, "Tutorial: Bullseye," and follow the tutorial there.

3 Tutorial: Hello World